

FIG. 1

```
public interface ViewAPI
{
11-1  → public ViewElement createElement(ViewType viewType, String name);
11-2  → public void removeViewElement(String name);
11-3  → public ViewElement getViewElement(String name);
}

public interface ViewElement
{
11-4  → public ViewProperty getViewProperty(String name);
11-5  → public void setViewProperty(String name, PropertyValue value);
11-6  → public void registerController
        (Controller controller, ControllerArguments args, UserInteraction event);
}
```

11

FIG. 2

```
public interface ModelAPI
{
12-1  → public ModelHome locateHome (String homeName);
}

public interface ModelHome
{
12-2  → public ModelElement createModelElement (ModelElementKey key);
12-3  → public void removeModelElement (ModelElementKey key);
12-4  → public ModelElementSet Query (QueryCriteria criteria);
}

public interface ModelElement
{
12-5  → public ModelElementKey getKey();
}
```

12

FIG. 3

```

// Controller

30-0  → public void ProcessSearchRequest
      {
        // access the view components
30-10  → TextField  inputName = ViewAPI.getViewElement(inputKey);
30-20  → String    employeeName = inputName.getProperty("data");

        // access the model
30-30  → EmployeeHome employeeHome = ModelAPI.locateHome("employees");
        // query the model
30-40  → Employee[] results =
        employeeHome.Query (new QueryCriteria ("findByName", employeeName));

        // generate the next view, and return to client
30-50  → setView(BuildSearchResultsView (name, results));
      }

      // View Generation code

30-60  → public void CreateInitialView ()
      {
        // create the top-level frame
30-70  → Frame    frame = ViewAPI.createViewElement (frame, "frameKey");
30-80  → TextField inputName = ViewAPI.createViewElement ("textfield", inputKey);

        // associate a controller with user-input into the textfield
30-90  → inputName.registerController (ProcessSearchRequest, null, TextField.CHANGE);
      }

```

FIG. 4

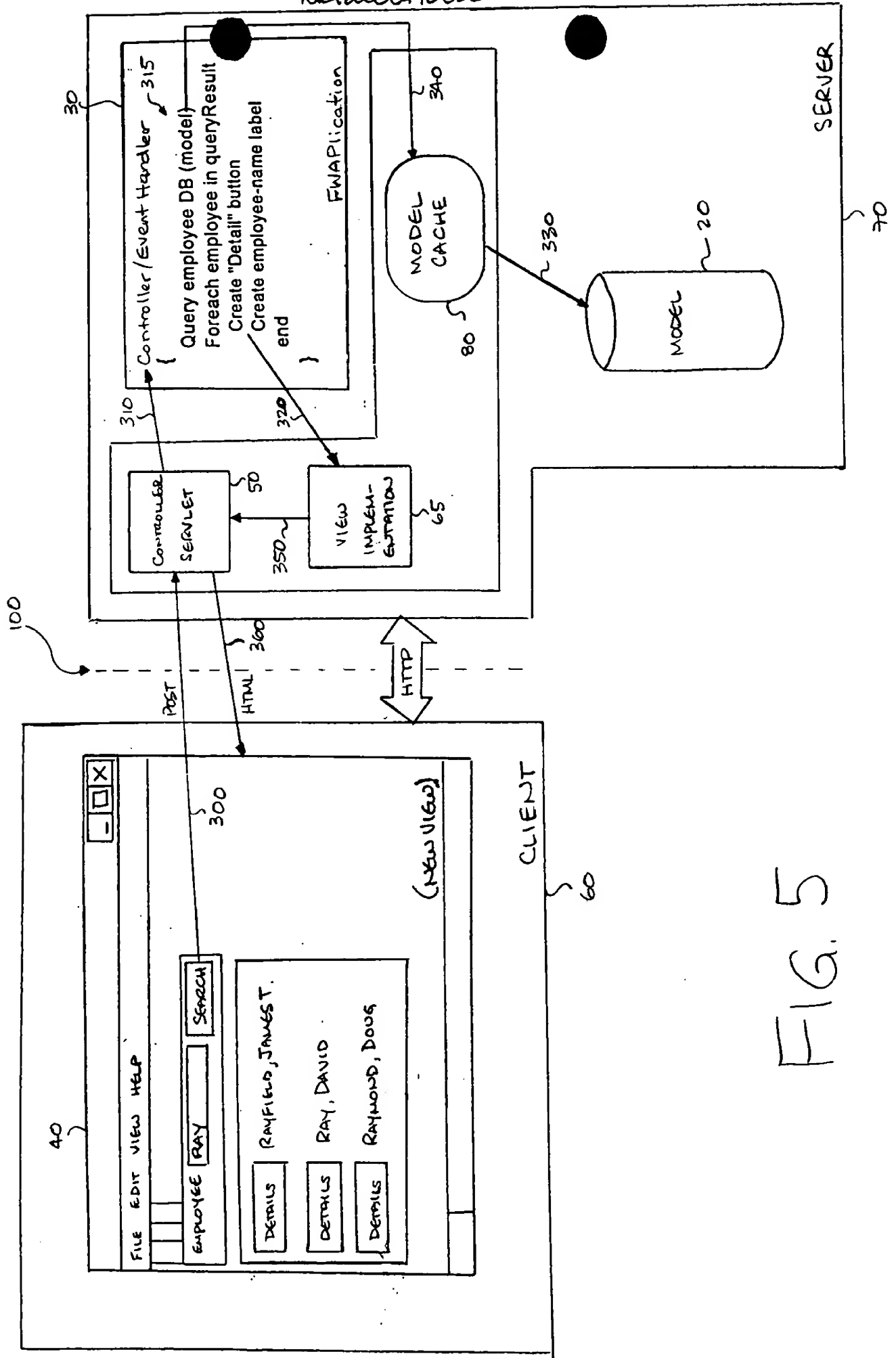


FIG. 6

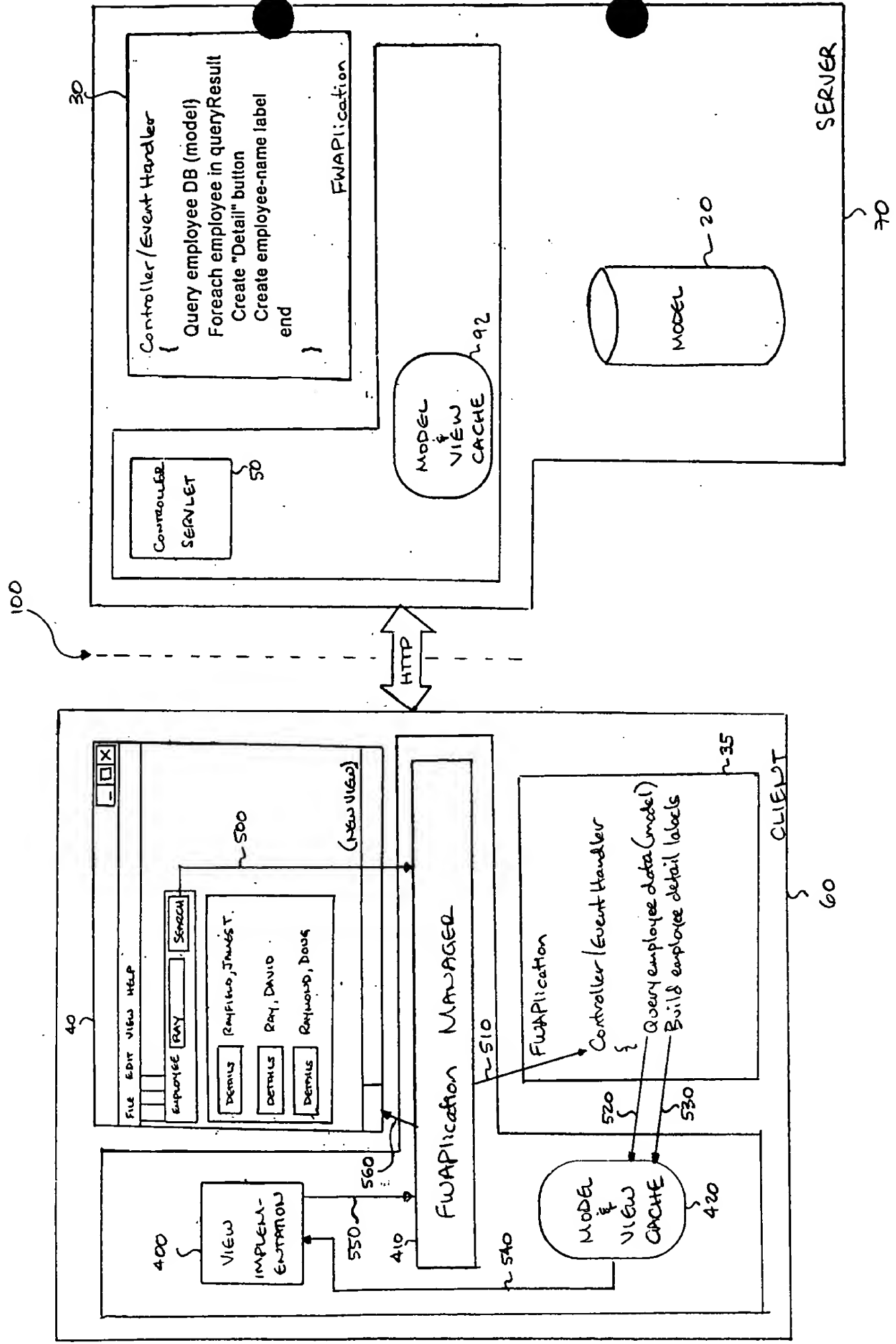


FIG. 6

FIG. 7 is a block diagram of a system architecture for a client-server application.

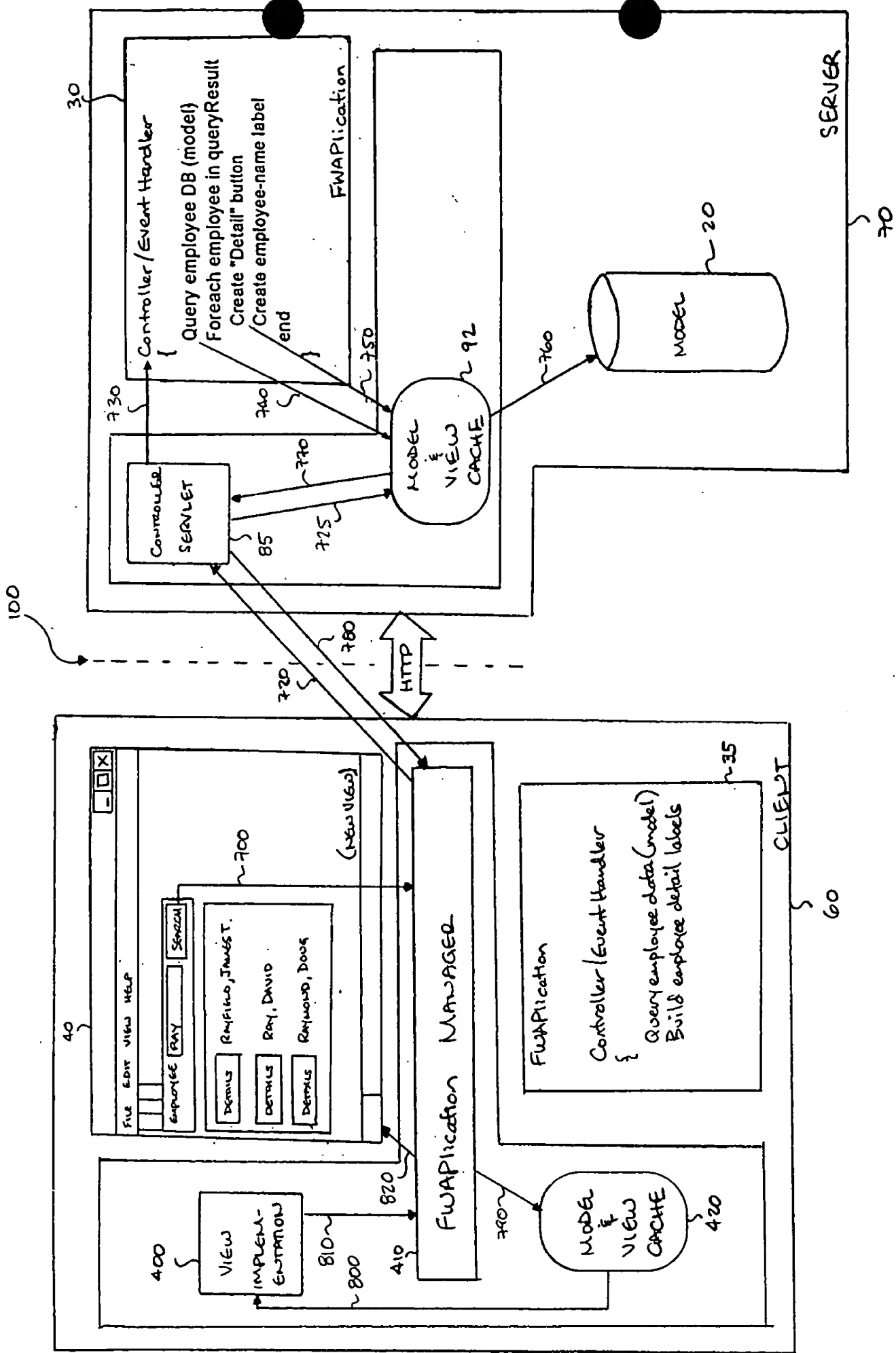


FIG. 7

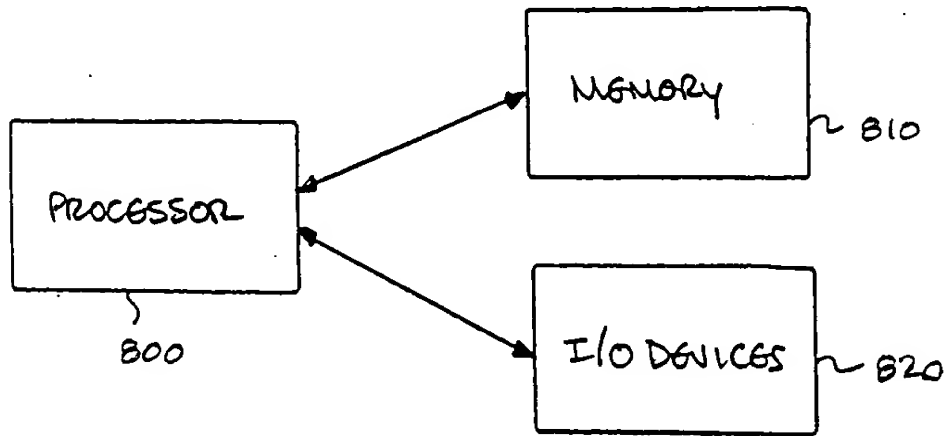


FIG. 8